



TITLE:

オートマタ理論における解析,制御  
及び合成の各志向の明確化につい  
て (数理情報科学の基礎理論と応用  
)

AUTHOR(S):

安在, 弘幸

---

CITATION:

安在, 弘幸. オートマタ理論における解析,制御及び合成の各志向の明確化について (数理情報科学の基礎理論と応用). 数理解析研究所講究録 1981, 421: 77-91

ISSUE DATE:

1981-03

URL:

<http://hdl.handle.net/2433/102549>

RIGHT:

## オートマタ理論における解析, 制御 及び合成の各志向の明確化について

九工大 安在弘幸

1. はじめに     オートマタ理論はいまや極めて膨大な研究  
研究成果を積みあげてきている。そして、  
情報工学からは、その基礎理論の1つとして熱い期待を寄せ  
られている。しかし、実際に膨大なのは、オートマトンのた  
めのオートマトンの研究であつて、情報工学の基礎として教  
えておく必要がある部分ははるかに少く、更に実践的な道具  
としてよりもむしろ、情報工学者にとって思考のための概念  
として重要なのだ、とも言われている<sup>1)</sup>。

確かに、例えば電気工学に対する電磁気学や電気回路論の  
ような、電気工学者に対してその Identity を与える基礎学の  
それらに比較し、情報工学に対する情報数学やオートマタ理  
論は、その基礎学としてはいかにも弱く、情報工学根無し草  
論が言及されても仕方がないようである<sup>1)</sup>。

しかし、情報工学にとって、電気工学における電磁気学や

電気回路論のような、その基礎学は必要がないというのではない。逆に、その必要性は益々増大している。

実際、電子計算機の原価の90%を占めるとさえ云われるソフトウェアの作成は、未だに手工業的であり、ソフトウェア危機と呼ばれる事態さえ云々されている。これを解決するためには、その作成工程の標準化や自動化など、その工業化を計らねばならない。

標準化という点に関して、Dijkstraはかの有名な“構造化プログラミング”と名付けた方法論を提案し、一挙に普及させた。比喩的に云えば、Dijkstraは情報工学におけるアダムスミスである。彼は、プログラミングという手作業工程に、分業と流水作業と、それを可能にする標準化の概念を持ちこんだ。こうして、ソフトウェアの生産過程に近代的手工業の方法論が導入されたのである。

しかし、電子技術の驚異的な発達に伴うハードウェアコストの低減は、ソフトウェアに対する需要を膨大かつ高度なものにし、その生産性や信頼性に対する要求を一段と強いものにしていく。産業革命時、工学は産業と互いに影響しあい、支えあいながら、マニファクチャを近代工業へと脱皮させていった。ところで情報工学は果して、上述のような要請に応え、ソフトウェアの生産に対し、その自動化などといった

近代工業化への方法を提供しうるであろうか。

提供しうるようになる, そして, その一翼を担うのが, データに対する操作の学としてのオートマタ理論である, と筆者は考えている。しかし, そのためには, 種々のオートマタ理論に対して, 標題に示したように, 解析, 制御及び合成の各々の志向の, どれに属するかを明確に区別して意識し, 且つそれぞれの志向に基く価値観のもとで, それぞれを展開させてゆく必要があるが, なかでも特に合成(方法)論的な理論を発展させてゆく必要がある。本稿では, その必要性とそれがどのようなものかについて議論する。

## 2. 認識論的価値観と方法論的価値観

一般に, 学問  
には理学的な

ものと工学的なものがあつてよく対比される。理学では, 我々の知的好奇心に触発されて, 自然的対象に対する「what …」または「why …」を追求し, その性質やそこに内在する法則に関する知見を増やす。一方, 工学では我々の生活環境を快適にするために「How …」を追求し, そこで得られた「How to …」を我々の知的財産とする。すなわち理学における研究は認識(存在, 解析)論的価値観のもとで行われ, 工学における研究は方法論的価値観のもとで行われる。

### 3. 制御論的立場と合成論的立場

工学では、対象へのアプローチの仕

方に、制御論的立場と合成論的立場がある。

制御論的立場では、与えられた対象を我々の目的に沿うよう変化（動作、変換または成長）させるため、出力（反応）を見つつ入力（刺激）を加える、すなわち対象を観測・制御する方法についての研究が行われる。この立場では、その対象に関する暗箱アプローチ（black box approach）的な情報すなわちその入出力関係が調べられる。また、その等価及至近似なモデルが作られ、利用される。その内部構造は、この暗箱アプローチ的な情報との関連において、推測され、仮定され、訂正され、改良され、そして検証される。

合成論的立場では、与えられた部品や材料としての対象を要素として、我々の目的に沿うように、人工物としての対象を合成する（組立てる）。この場合、部品としての対象に関する知見については、暗箱アプローチ的な表現＝入出力関係記述、すなわち制御論的立場からの知見であればよい。しかしこれらの部品から組立てられる対象については、これらの部品としての対象が互いに結合して目的の対象を構成する、その結合の仕方およびその効果という意味での、内部構造に関する情報を十分に知っておかなければならない。この場合、

その結合の仕方は、それら部品(要素)の相互の物體的な或いは入出力的な結合による。このとき、このように構成としてとらえられる対象はシステムと呼ばれる。

それぞれの学問は、以上述べた3つの立場のどれかに主として依拠しているようである。その例を次に示す。

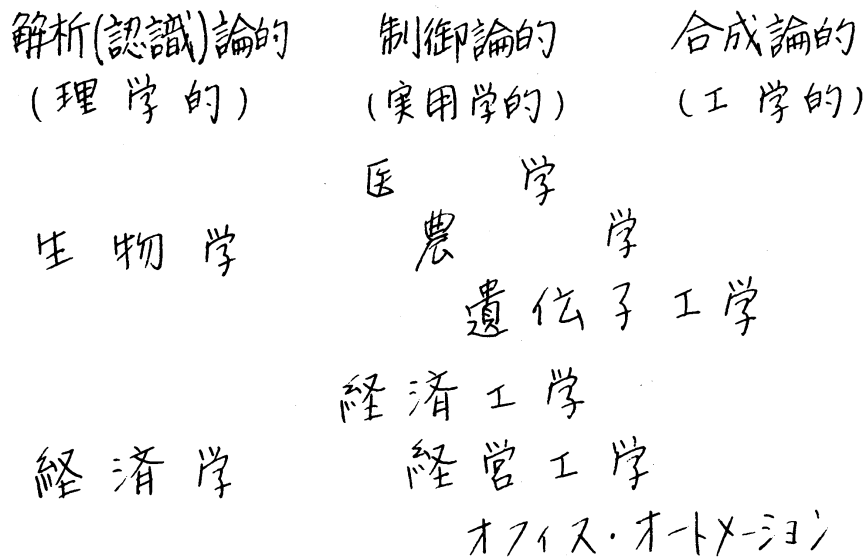


図1. 学問領域と解析, 制御及び合成の各々の志向

#### 4. Physical と Logical

計算機システムにおいて、

Physical と Logical という

対比概念がよく使用される。回路設計者にとっては、トランジスタやキャパシタなどが Physical で、フリップフロップやアンド素子などが Logical である。更に論理設計者にとっては、そのフリップフロップやアンド素子などが Physical で

あり、レジスタやカウンタなどが Logical である。また、O.S. を Logical とする立場ではハードウェアシステムは Physical であり、一般ユーザにとっては O.S. は Physical である。計算機システムはこのような Physical と Logical の多層な階層をなしている。

同様なことが学問の対象領域についても言える。例えば、鉄板は、機械(設計)工学者にとっては Physical であり、物理(物性)学者や冶金工学者にとっては Logical である。

こうして我々は、我々の対象を Physical と Logical の階層として見、理解しようとしていることが分る。明らかに、上層の Physical は下層の Physical から組立てられたシステムと見なされ、そしてこのシステムが Logical と呼ばれている。以後、この上層及び下層の Physical とそれぞれ表相及び深相 Physical と呼び、そのシステムを組成 Logical と呼ぶ。

解析(認識)論では、Physical とみなしている対象の内部に、深相 Physical とその上に構築された組成 Logical (system) と見ようとする。或いは逆に、与えられた Physical の上に構築された組成 Logical (system) について、それがどんな表相 Physical に見えるかについて知ろうとする。

制御論では、与えられた対象を表相 Physical と見るのだが、その際その下に見ている Logical は仮想的、すなわちそれに

等価及至近似なモデルであつてよい。ここでは、その対象の自律性に大きく依存している。

合成論では、対象に関する要求仕様が、暗箱アプローチ的表現=入出力関係記述として与えられる。そして次に、それを満足するように、与えられた部品対象を組立てて、目的の対象を構成しようとする。換言すれば、記述として与えられた表相Physicalを、深相Physicalを要素とする組成Logical (system) として与えることにより、実体としての表相Physical にしようとし、そのための方法を研究する。

## 5. オートマタ理論

以上の議論をオートマタ理論について考察する。解析(認識)論

的オートマタ理論では、あるPhysical (を深相とし、その)の上に構成されたLogicalについて、その構成の仕方の様態に応じ、それを組成Logicalとする表相Physicalの性質が調べられる。次にその記述の仕方を述べる。まず、深相Physicalの記述が与えられる。それから、それを基にbottom-up的にLogicalが組立てられ、その暗箱的表現=入出力関係記述として、表相Physicalの記述が与えられる。ここで、その記述の仕方の方向は総合的(synthetic)である。

この理論ではまた、表相Physicalが与えられたとき、その



内部を組成 Logical と深相 Physical に分解するように分析する。この場合、組成 Logical は解析モデルを与える。これは制御論の立場と重なる。一般に、自動制御論や現代制御論では、その対象である表相 Physical は自然的 (連続量表現) で自律的であるのに、オートマタ理論では人工的 (離散量表現) で他律的である。

合成論的なオートマタ理論では、通常の合成論とまったく同様に、まず要求仕様が、合成しようとする目的のオートマトンの入出力関係記述 = 暗箱的表現として与えられる。こうして、表相 Physical の記述が与えられると、これを深相 Physical である部分対象を要素として組成 Logical を組立て、これを与えられた記述を満足する表相 Physical として実体化する、その方法を与える。このとき、最初にとえられた表相 Physical の記述は、深相 Physical の記述と組成 Logical の記述に、top-down 的に分解されて記述される。この記述の仕方の方向は明らかに分析的 (analytic) である。

#### 6. 人間向きな手順と機械向きな手順

一般に、方法とか手順さ

かえたもの、すなわち与えられた対象 (データ) に何らかの目的を果すように加えられる操作の手順は、人間向きと機

械向きとでは、それぞれの特性の相違に応じて、それぞれ異なる筈である。このことは、手作業の自動(機械)化、手工業の近代工業化などといった場合に、当然に検討しておかなければならない問題である。

その相違とは、例えば因数分解のように、人間には簡単でも計算機には大変であるようなものである。これは、人間がその洞察力、パターン認識力、学習能力などを使い、いわゆる発見的に問題を解くのに優れた能力を有しているのに、現在の計算機ではこの種の能力が本来的な意味で欠除しているからである。従って、人間に適した方法(解法)が、そのまゝ計算機に適した方法(解法)になるとは限らないのである。

このような発見的方法に対して、与えられた問題もある定められた手順に従い機械的に解いてゆく方法があり、算法(algorithm)と呼ばれる。計算機はこの算法を実行する機械である。ところで、人間が実行する算法(と呼ばれているもの)は古くからある。しかしこれが果して計算機に適したものになるとは限らない。

我々は、経験的に次の事実を知っている。人間は計算手順があまりに機械的になるのは好まない。また、計算速度も遅い。そこで適当に学習し、洞察することによって処理過程の経済を計り、手数を減らそうとする。人間の算法には、かな

り人間の洞察力に依存するもの、大味なもの、またはパターン認識力に頼っているものがある。

一方、現在の計算機にとって真に好ましい算法は、局所化された対象に、少種の単純な操作や判断を繰返し適用し、それによって目的の処理が遂行されるような種類のものである。こうして、次の命題が了解されよう。すなわち機械的な方法と云っても、人間向きの方法が直ちに機械向きの方法として適しているとは限らない。

例えば、線形連立方程式を解く問題を考えればよい。数値解析では、CRAMERの公式は顧みられない。一方 Gauss-Seidel 法も筆算向きに勧めるものはいない。消去法は両者で使われる。しかし人間向きには無意識的に洞察力に頼る発見的な方法が部分的に使われ、一方機械向きでは完全に組織的で、決りきったことも近道はせず、手数を厭わず、馬鹿丁寧な計算を進める。つまり、機械の特性は人間の特性とはかなり違っているから、それらに適用される方法は、それぞれの特性に応じて異なるのが当然であろう。このことは数値処理ではよく研究もこれ理解も進んでいるのに、記号列処理では一般の理解は未だしの感がある。従ってこれらの処理についての研究では次の方法論的要請を常に心しておくべきであろう。すなわち、人間には人間向きの、機械には機械向きの、それぞれ

の特性に応じた方法が必要である。

因数分解などのように、人間向きな記号列処理が既に存在していたのとは異って、コンパイラやそれに関連した分野では、機械向きな記号列処理として独自に各種の実際的な技法を開発し、その理論に進められた。このことは、機械向きな方法論と人間向きな方法論から独立させ、機械向きということの原理に従って構築させねばならないことを示している。

## 7. 処理手順の開発

オートマタ理論, アルゴリズム論, 帰納的関数論など, 機械的

な処理手順の存在に関する研究は既に膨大である。また処理手順の測度として、その複雑さに関する研究も始まっている。ところで、これらの処理手順の開発に関する研究はどのようなになっているであろうか。処理手順の開発、すなわちソフトウェアの作成、更にはより具体的にプログラムの作成といった作業は、次の性質がある。すなわち、そこで開発される対象は、機械向きの手続きであるが、その作業すなわち開発処理自体は、現在のところ人間向きである。

Dijkstraはこの手作業工程に、構造化プログラミングという近代的手工業の概念を持ちこんだ。ここでは、表相Physicalのレベルの記述からtop-down的に深相Physicalと組成Logi-

cal のレベルの記述へと書き直される。そして必要があれば、深相Physicalの要素が再び浅相Physicalとしてとらえられ、上述の過程が繰返される。この場合、この深相Physicalにおける各要素は互いに直接的な連結や干渉はなく、ただ組成Logicalとのみ本構造的に連結されるように分解される。

ところで既に述べたように、現在情報工学に課せられた問題は、この処理手順の開発処理手順を自動化し、機械向きにすることである。換言すれば、処理手順の実行機械を自動合成する機械を与えることである。このためには、明らかに合成論志向のオートマタ理論が必要である。

さて、この種の処理手順の開発処理手順の自動化(機械化)の問題の典型に、Compiler・Compilerの問題がある。ベルギーのLeuven大学の言語工学研究所のLewiらは、Siemens社(ベルギー支社)とコンパイラ作成用ツールLILAを共同開発しているが、そこで採用している基本概念は合成論的なオートマタ理論の原理である<sup>2)</sup>。一方、筆者らもこの十年来、同様な立場からの研究を続け、古典的オートマタ理論を合成論的立場から再編成し<sup>6)</sup>、それを基礎にその延長として、再帰降下コンパイラ(の中心機構をなすtranslator)の自動生成アルゴリズムを与え、その正当性を証明し、それを実行する生成機械を与えることが出来た<sup>3), 4)</sup>。そしてこの原理に基づいて、

コンパイラやインタプリターの自動生成システム LPG を試作し、幾つかの言語処理系を生成させ、その実効性を確かめた<sup>5)</sup>。

## 8. 構造化オートマタ理論

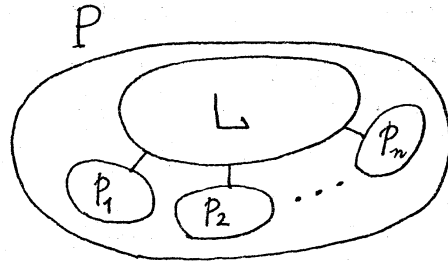


図2. Physical と Logical

表相 Physical を  $P$ ，深相 Physical を  $p_i$ ，組成 Logical を  $L$  と表すと，これらの関係は図2のように表すことができる。いま，この表相 Physical  $P$  を言語処理機械とみなすとき，深相 Physical  $p_i$  は部品機構（部分機械），組成 Logical  $L$  は制御機械とみることもできる。いま， $L$  を順序機械， $p_i$  を操作機構付きデータ構造（これは SIMULA67 の class に当る）とする。そして例えば， $p_1$  が push-down stack であるとする。このとき，従来のオートマタ理論ではこれを直ちに push down stack automaton と呼ぶ。このアプローチは解析論的であり，理学的である（と我々は考える）。その理由は3.及び4.に示した通りである。

一方，筆者や Lewi らは，これを機械  $L$  の拡張や一般化ともみなす<sup>5), 2)</sup>。このとき push-down stack  $p_1$  は， $L$  により制御される単なる部品機構（深相 Physical の要素）に過ぎないとみなされる。このような考え方により取扱れる機械  $P$  を

筆者は広義のL機械と呼んだのだが<sup>5)</sup>, Lewiらはこれを一般化 (Generalized) L機械と名付けている<sup>2)</sup>。このアプローチが合成論的であり, (組立)工学的であることも既に3.及び4.で述べたとおりである。

ところで, 解析論的立場の記述では, 部分の記述から出発して, bottom-up的に全体を統合する記述に到る, 総合的な方向をとる。これに対し, 合成論的立場の記述では, 全体の記述から出発して, top-down的に個々の部分の記述へと進む, 分析的な方向をとる。その際, その部分の分割の進め方と, その各部分がそれを統括する機構と木構造的に連結されるようにしてゆくのが, Dijkstraのいわゆる構造化の考え方であった。従って, ここではこのような考え方に基く合成論的な理論を構造化オートマタ理論と呼ぶことにする。

この合成論では, 部品機構  $p_i$  はあらかじめ与えられた完成品とみなされ, それらを制御する機械  $L$  の合成の方法が問題になる。すなわち, この機械  $L$  の合成工程に自動化が導入される。この方式では少しいのコストで  $L$  を再合成できるので, 部品  $p_i$  の取替えは容易であり, 従って機械  $P$  を試行錯誤的に, すなわち実際の使用環境に適合させるように調整 (tuning) させながら容易に作成 (implement) できる<sup>3), 4), 5)</sup>。

これは、効率的で優れたシステム作成方式であろう。

## 9. むすび

以上, 情報工学の基礎として, オートマ  
タ理論は解析, 制御及び合成のそれぞれ  
の志向に基く価値観のもとでそれぞれを発展させる必要があ  
ること, なかでも合成論的立場が重要であることを示した。  
筆者らは, 情報工学に対するオートマタ理論の関係を次のよ  
うにとらえ, 半(環上の)線形代数を現在展開中である。<sup>6)</sup>

電気工学 ... 電磁気学 ... 電気回路論 ... 線形代数

情報工学 ... 解析論的オートマタ理論 ... 合成論的オートマタ理論 ... 半線形代数。

## 参考文献

- 1). 坂井他編: "情報工学の教育・研究", 5章, bit臨時増刊, 12, 共立(1980).
- 2). Lewi, J., et al.: "A Programming Methodology in Compiler Construction, Part 1: Concepts", North-Holland (1979).
- 3). 安在, 潮崎: "再帰降下順序変換機系とその生成機械", 信学論(D), J63-D, 9, pp. 771-8 (1980-09).
- 4). Anzai, H.: "A theory of recursive descent translator generator", Proc. of Int'l. Comp. Symp. ICS80 (1980-12).
- 5). 安在他: "言語処理系の自動生成システム LPG/I について", 信学技報 AL79-66 (1979-11).
- 6). 安在他: "半線形代数" 第1報, 第2報, 第3報, 信学技報 AL80-60, 61, 62 (1981-01).